UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/183,694	10/30/1998	JACKSON L. ELLIS	98-179	3415

24319 7590 05/20/2005

LSI LOGIC CORPORATION
1621 BARBER LANE
MS: D-106
MILPITAS, CA 95035

EXAMINER	
PARK, ILWOO	
ART UNIT	**PAPER NUMBER**
2182	

DATE MAILED: 05/20/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/183,694
Filing Date: October 30, 1998
Appellant(s): ELLIS ET AL.

Christopher P. Mariorana
For Appellant

EXAMINER'S ANSWER

Responsive to the reply brief filed on 5/23/2003, a supplemental Examiner's Answer is set forth below:

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Invention

The summary of invention contained in the brief is deficient because it only supports for a portion of claims.

'Re-ordering' may be found in page 3, lines 23-26, page 4, lines 1-3, page 8, lines 20-22, page 40, lines 27-28, page 41, lines 5-7, page 43, lines 15-17, page 44, lines 15-17, page 45, lines 1-3, and page 46, lines 18-19.

'Generating interrupts at beginning and at end of the plurality of commands' would be found in page 3, lines 17-22.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(8) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

(9) Prior Art of Record

5,781,803	KRAKIRIAN	7-1998
5,483,641	JONES et al	1-1996
4,543,626	BEAN et al	9-1985

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim 21, 22, and 26 are rejected under 35 U.S.C. 102(e) as being anticipated by Krakirian, US patent No. 5,781,803.

As to claim 21, Krakirian teaches a data controller [hard disk controller IC 204] of a peripheral device [target device 202 in figs. 3 and 4; col. 7, lines 7-24] having a storage medium [hard disk 208] and a processor [microprocessor 206], wherein the data controller data controller minimizes interrupts [elimination of an interrupt to the microprocessor for seeking operation in case that reordered commands parsed by the controller are contiguous disk block accesses each other: col. 3, lines 38-42; col. 16, line 34-col. 17, line 5; col. 17, lines 36-63] to the processor by re-ordering [col. 15, lines 13-28] a plurality of commands received from a host computer [initiator 201] from an order of arrival into an order of sequence in the storage medium.

As to claim 22, Krakirian teaches a command queuing engine configured to arrange the plurality of commands into at least one thread [col. 17, lines 40-51].

As to claim 26, Krakirian teaches a peripheral device [target device 202 in figs. 3 and 4; col. 7, lines 7-24] that includes a data controller [hard disk controller IC 204], microprocessor [206], a buffer memory [205], local memory and a storage medium [hard disk 208], and that is couplable to a host [initiator 201], wherein the data controller creates [col. 17, lines 40-63] threads of a plurality of commands and generates interrupts [col. 4, lines 4-43] at the beginning and end of the of the plurality of commands relative to a data transfer.

Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Krakirian, US patent No. 5,781,803 and Jones et al., US patent No. 5,483,641.

As to claim 3, Krakirian teaches a data controller [hard disk controller IC 204 in figs. 3 and 4], that is couplable to a host [initiator 201] and coupled to a storage medium, microprocessor, local storage and a buffer memory, comprising a command queuing engine that creates a plurality of threads of sequential commands [col. 15, lines 18-27; col. 17, lines 36-63] while minimizing [col. 5, lines 8-12] interrupts associated to the commands. Even though Krakirian teaches creating a plurality of threads of sequential commands, Krakirian does not explicitly disclose the plurality of threads of sequential commands exist simultaneously.

Jones et al teach a data controller queues a plurality of commands, reorders the commands, and creates a plurality of threads of sequential commands [multiple sequential read or write requests; a plurality of COMB-ORIGs] that exist simultaneously [col. 6, lines 10-21; col. 50, lines 50-60; col. 53, lines 1-63].

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Krakirian and Jones et al because they both teach holding a plurality of commands in a command queue, reordering the commands for sequential accesses, and creating threads for accesses of a hard disk drive and the Jones et al's teaching of the plurality of threads of sequential commands existing simultaneously in a queue would increase efficiency of seek operation by reordering of a plurality of commands and making threads as many as possible for the commands in the queue of Krakirian.

Claims 16-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krakirian, US patent No. 5,781,803 and Jones et al., US patent No. 5,483,641 as applied to claim 3 above, and further in view of Bean et al., US patent No. 4,543,626.

As to claims 16 and 18, Bean et al teach a command queuing engine comprises:

a transfer extend generator configured to generate [col. 4, lines 38-42] transfer extend entries for a data transfer between the storage medium and a host computer; and

a data retrieval channel [col. 6, lines 64-68] coupled to receive the transfer extend entries for programming the data transfer.

As to claim 19, Bean et al teach the command queuing engine further comprising a status retrieval channel [col. 3, lines 59-64; col. 4, lines 18-21].

As to claim 20, Bean et al teach each of the retrieval channels are coupled to receive transfer extend entries and to provide [implicit: col. 7, lines 29-57] used read pointers to a first storage device of the peripheral device.

As to claim 17, Bean et al teach the transfer extend generator is coupled to the buffer memory to store the transfer extend entries [col. 4, lines 15-18].

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Krakirian, Jones et al, and Bean et al because they both teach a data controller [Bean et al: e.g., host interface controller processor 12 and source processor 18 in fig. 1] for receiving commands from a host and a microprocessor [Bean et al: e.g., disk controller processor 14 in fig. 1] for performing hard disk access operations in accordance with the host commands received and queued and the Bean et al's teaching of generating transfer extend entries from the host commands would increase efficiency by further reducing interrupts [col. 12, lines 32-43 of Krakirian and col. 2, lines 3-17 and col. 6, lines 23-32 of Bean et al] to the microprocessor of Krakirian and Jones et al.

Claims 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krakirian, US patent No. 5,781,803 as applied to claim 21 above, and further in view of Bean et al., US patent No. 4,543,626.

As to claims 23-25, Bean et al teach the limitations of the claimed invention [vide supra].

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Krakirian and Bean et al because they both teach a data controller [Bean et al: e.g., host interface controller processor 12 and source processor 18 in fig. 1] for receiving commands from a host and a processor [Bean et al: e.g., disk controller processor 14 in fig. 1] for performing hard

disk access operations in accordance with the host commands received and queued and the Bean et al's teaching of generating transfer extend entries from the host commands would increase efficiency by further reducing interrupts [col. 12, lines 32-43 of Krakirian and col. 2, lines 3-17 and col. 6, lines 23-32 of Bean et al] to the processor of Krakirian.

(11) Response to Argument

The Examiner summarizes the various points, which are not addressed in the Grounds of Rejection above, raised by the Appellants and addresses replies individually. Further, the Krakirian reference is better understandable as a whole not a specific portion to where the Examiner pointed.

The Appellants argue in substance that **a)** regarding claim 21, Krakirian shows reordering function but does not disclose that the reordering is performed by a data controller (page 6, line 11-page 7, line 3 in Appeal Brief), Krakirian does not show which element among a hard disk controller IC and a microprocessor provides the reordering function that a data controller reorders a plurality of commands (page 6, line 11-page 7, line 3 in Appeal Brief) while the language, "wherein the data controller minimizes interrupts to the processor by re-ordering a plurality of commands" of claim 21 is clear (page 3, lines 2-4 in Reply Brief) and none of Krakirian mentions the hard disk controller 204 and Krakirian shows the CFIFO 217 is part of the SCSI interface portion 211 and not the hard disk controller 204 (page 4 in Reply Brief), **b)** regarding claim 21, Krakirian discloses not minimizing interrupts rather shows that the microprocessor 206 is interrupted upon receipt of each command (item 3 in pages 7-8 of Appeal Brief, page 4

in Reply Brief) , **c)** regarding claim 26, Krakirian does not disclose or suggest a data controller creating a plurality of threads of a plurality of commands (item 4 in pages 6-7 of Appeal Brief, page 5 in Reply Brief), **d)** regarding claim 26, the queue of Krakirian cannot hold plural threads simultaneously (item 5 in page 10 of Appeal Brief), and **e)** regarding claim 3, no motivation to modify Krakirian with Jones et al (item 6 in page 13 of Appeal Brief),

For the point a), firstly, the language of claim 21 is not clear; the language does not clearly say that the data controller performs reordering; the language does not necessarily mean the data controller performs reordering; contrarily, as supported in Specification of the Appellants' application describing:

"By defaults, CQE handles commands in the order they arrive. However, the **firmware** receives the **interrupts** required to allow **re-ordering** of the commands into sequential threads or for the purpose of reducing disk seek time (page 3, lines 23-26)" or

"The **microprocessor** may also wish to determine if the third (write) thread is sequential to the first (write) thread and **re-order** the second read to occur after the third write (page 44, lines 15-17)",

re-ordering is performed by the microprocessor; thus, claim 21 does not require re-ordering done by either a data controller or a processor and requires the existence of the function 're-ordering'; the function 're-ordering' is shown by the Krakirian reference and recognized by the Appellants. Further, the 're-ordering' of the Krakirian reference is done in the CFIFO 217 which is part of the SCSI interface portion 211 which is part of the hard disk controller IC 204 [see figs. 3-4].

For the point b), Krakirian teaches receiving a burst of commands and re-ordering commands having logical block addresses to be sequentially rearranged [e.g., from outer track to inner track of disk in col. 15, lines 13-27]. Each one of commands

needs a seek operation which requires intervention of a microprocessor [col. 3, lines 38-42]. However, if next command which is re-ordered and resulted in a starting disk block address sequential to an ending disk block address of the one command [e.g., col. 17, lines 36-63; col. 16, lines 47-52], then at least one interrupt to the microprocessor is minimized because another seeking operation for the next command is not needed **or** at least one interrupt to the microprocessor is minimized because the next command is determined to be an ESP command which does not require a microprocessor intervention [col. 12, lines 58-67]. And the portion that the microprocessor 206 is interrupted upon receipt of each command of Krakirian does not mean adding an interrupt; rather, Krakirian now is able to **eliminate** the interrupt [col. 12, lines 58-67] which was always necessary in prior art of Krakirian [col. 4, lines 19-23].

For the point c), Krakirian teaches examples of creation of a plurality of threads of a plurality of commands: **one thread** [col. 12, lines 58-67] with two commands for execution without microprocessor intervention between the two commands and **another thread** [col. 17, lines 51-63] with two commands for execution at once from buffer memory to one area of disk without intermittent seeking.

For the point d), claim 26 does **not** have a limitation of 'a **queue holding** plural threads **simultaneously**.'

And for the point e), normally, one thread is for executing one command; instead of being two threads for executing two commands, Krakirian teaches one thread created for executing two commands [col. 12, lines 58-67; col. 17, lines 51-63]. Krakirian teaches receiving a plurality of commands, holding the plurality of commands in a

command queue and reordering the commands to create a thread of sequential commands for sequential accesses; though Krakirian has a queue, Krakirian does not expressly disclose multiple threads existing at one time; in other words, Krakirian shows only one thread exists at one time; thus, commands received can be analyzed/determined to be executable with the one thread or not and the one thread is being in disk operation currently or the one thread is just about to be in disk operation. And Jones et al teach a queue holding a plurality of commands in a command queue and reordering the commands to create threads that exists simultaneously in the queue; in other words, the queue of Jones et al holding multiple threads enables at least one thread of the multiple threads could be used for the commands analysis/determination and consequently increases efficiency of disk seek operation. Further, the queue of Jones et al holding multiple threads increases reliability [col. 50, lines 55-57 in Jones et al] to simplify error handling in the queue. Therefore, one of ordinary skill in the art would be easily motivated to modify the queue of Krakirian useful to hold multiple threads as taught by Jones et al.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

ILWOO PARK
PRIMARY EXAMINER

Ilwoo Park
May 11, 2005

Appellant may file another reply brief in compliance with 37 CFR 41.41 within two months of the date of mailing of this supplemental examiner's answer. Extensions of time under 37 CFR 1.136(a) are not applicable to this two month time period. See 37 CFR 41.43(b)-(c).